PROCEEDINGS OF SCIENCE

ALISE - The Account LInking SErvice

Dr. Marcus Hardt[a,*] **Dr. Diana Gudu**[a] **and Paul Millar**[b]

[a] *Karlsruhe Institute of Technology,*
Herrmann-von-Helmholtz-Platz 1, 76344 Eggenstein-Leopoldshafen, Germany
[b] *Deutsches Elektronen-Synchrotron DESY,*
Notkestraße 85, 22607 Hamburg, Germany

E-mail: hardt@kit.edu, gudu@kit.edu, paul.millar@desy.de

Modern research computing environments increasingly rely on federated identity management to provide seamless access to distributed resources. However, many computer centres face a significant challenge: they need to support federated users while maintaining compatibility with traditional Unix-based systems that require local account mappings. ALISE (Account LInking SErvice) addresses this gap by providing a robust solution that enables users to link their federated identities with local computer centre accounts.

ALISE is a comprehensive web application designed to serve three distinct user communities. End users can easily manage their account linkages through an intuitive web interface. Application developers can integrate federated identity mapping into their services via a well-defined REST API. System administrators can deploy and configure ALISE to implement institution-specific policies while maintaining operational simplicity.

Within the AARC Blueprint Architecture framework, ALISE operates at the "End-Services" layer, implementing account linking at the service level rather than at higher authentication layers. This design choice enables direct integration with existing infrastructure while maintaining security and governance requirements.

The service workflow begins when users authenticate with their existing computer centre accounts. Subsequently, they can link multiple federated identities to their local accounts, creating a comprehensive mapping database. Authorised services can then query this database through the ALISE REST API to resolve federated identities to their corresponding local Unix accounts, enabling seamless integration with traditional authorisation systems.

ALISE has been successfully deployed to support storage systems such as dCache and general-purpose services like teapot. The system is available as open source software under the MIT License, encouraging community adoption and contribution.

International Symposium on Grids and Clouds (ISGC2025)
16 - 21 March, 2025
Academia Sinica Computing Centre (ASGC), Institute of Physics, Academia Sinica Taipei, Taiwan

*Speaker

 ISSN 1824-8039 . Published by SISSA Medialab.

PoS(ISGC2025)012

1. Introduction

The landscape of research computing is increasingly embracing federated identity management, enabling researchers to access distributed computational and data resources using their institutional credentials. However, this evolution creates a significant integration challenge for computer centres that must bridge the gap between modern federated authentication systems and legacy Unix-based infrastructure that requires local account mappings.

The Account LInking SErvice (ALISE) [1] provides a comprehensive solution to this challenge by enabling seamless integration between federated identities and local computer centre accounts. ALISE addresses scenarios where remote users require access to services that operate on Unix-based systems, while preferring to authenticate using their federated credentials rather than managing multiple local accounts across different institutions.

The core challenge stems from the architectural mismatch between federated authentication systems, which provide dynamic identity assertions, and traditional Unix systems, which require static local user accounts for file system operations, process ownership, and access control. ALISE resolves this by maintaining persistent mappings between federated identities and local accounts, enabling services to seamlessly translate between these two identity domains.

ALISE is architected as a standalone web service providing two complementary interfaces. The user-facing web interface enables individuals to establish and manage their identity linkages through an intuitive workflow. The service-facing REST API allows applications and services to query the mapping database to resolve federated identities to their corresponding local accounts. This dual-interface design ensures that ALISE can serve both human users and automated systems effectively.

A concrete example illustrates ALISE's value proposition: consider a researcher uploading data to a computer centre via WebDAV [2]. The WebDAV service authenticates the user through an OIDC access token, establishing their federated identity. However, the service must store the uploaded files with appropriate Unix ownership so that the same user can later access them via SSH or other local tools. ALISE enables the WebDAV service to query the mapping database, translating the federated identity to the corresponding local Unix account, thereby maintaining consistent access control across different access modalities.

This paper provides a comprehensive examination of ALISE from multiple perspectives. We begin by contextualising our approach within existing account linking methodologies in section 2. The remainder of the paper is organised around the three primary stakeholder groups that interact with ALISE: end users (section 3), system administrators (section 4), and application developers (section 5). This structure ensures that each stakeholder community can understand how ALISE addresses their specific requirements and operational concerns.

2. Related Work

The idea of account linking is not new. Most approaches, however, relate to it when linking on a higher level in the AARC Blueprint Architecture [3]. See also Fig. 1.

The majority of existing account linking solutions operate at higher architectural layers, typically between Community AAI systems and Identity Providers. Common use cases include home

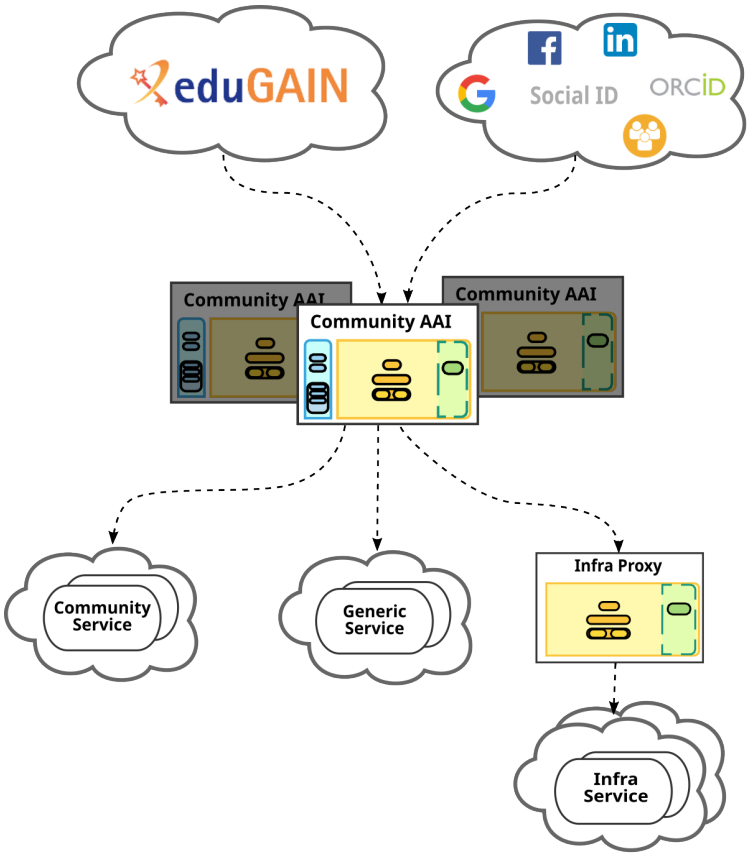

Figure 1: The AARC Blueprint Architecture 2019, showing the positioning of different account linking approaches. ALISE operates at the End-Services layer (highlighted region), linking federated identities directly to local accounts. Traditional linking approaches typically operate at higher layers between Community AAI and Identity Providers, focusing on attribute aggregation rather than local account mapping.

institutions or Community AAIs integrating external identifiers such as ORCID [4] into the attribute sets released to downstream services. Similarly, services may link to external providers to enhance assurance levels or obtain additional user attributes.

These higher-layer approaches typically involve Identity Providers or Community AAI systems offering web-based interfaces where users can initiate linking workflows. Upon successful authentication with the external service, additional attributes are retrieved and persistently associated with the user's primary identity. When assurance levels are involved, implementations must adhere to the AARC recommendations on Account Linking (AARC-G031) [5], which specify security and governance requirements for linking external identities.

A critical distinction exists between two fundamental account linking paradigms, both of which employ the term "login" but serve different purposes. The first paradigm uses the linking process primarily for attribute enrichment: users authenticate with external services to obtain additional attributes, but subsequent access must still occur through their primary identity. The second paradigm enables the linked identity itself to serve as an alternative authentication mechanism, allowing users to access services using any of their linked identities regardless of when the linking

was established.

Both paradigms must carefully address affiliation and status information, particularly in academic and research contexts where services need to determine a user's current institutional status. The AARC recommendation AARC-G025 [6] provides guidelines for expressing affiliation information in ways that enable services to make appropriate authorisation decisions based on current organisational relationships.

ALISE distinguishes itself by operating at the End-Services layer, focusing specifically on the challenge of mapping federated identities to local Unix accounts rather than attribute aggregation or alternative authentication pathways. This positioning enables ALISE to address the specific requirements of computer centres that must maintain compatibility with traditional Unix-based infrastructure while supporting modern federated authentication workflows.

3. User Experience: The Account Linking Process

The account linking workflow in ALISE is designed to provide users with an intuitive yet secure mechanism for establishing persistent mappings between their federated identities and local computer centre accounts. This section details the step-by-step process from a user's perspective, emphasizing both the simplicity of the interface and the robustness of the underlying security model.

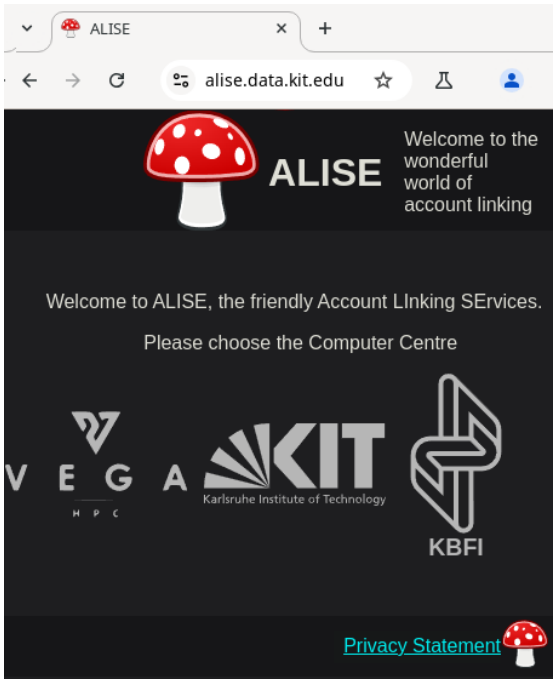

Figure 2: ALISE computer centre selection interface, where users choose their target institution for account linking

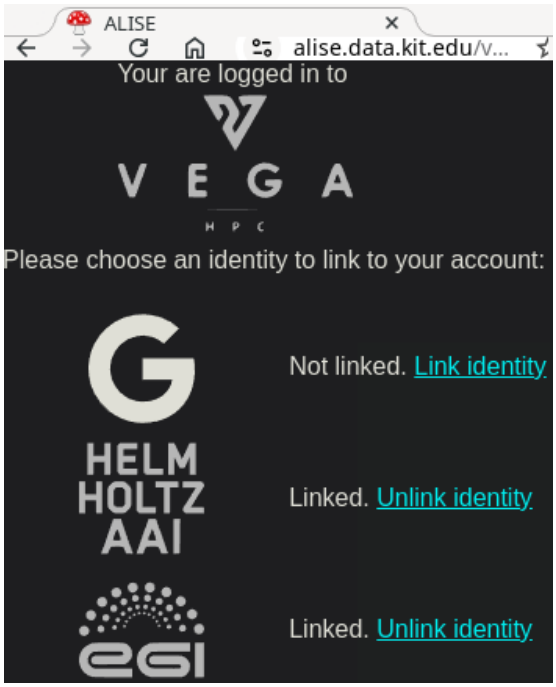

Figure 3: ALISE identity management interface, displaying available identity providers for linking and current linkage status

The account linking process begins with user authentication using their *internal* (primary) account, which represents their existing local account at the computing centre. This initial authen-

tication step is crucial for establishing a trusted foundation for the linking process, as it ensures that users can only link federated identities to accounts they legitimately control.

Since ALISE's primary use case involves mapping federated identities to Unix accounts, the local account's username must be included in the attributes provided during authentication. This design ensures strict adherence to local computer centre username policies and governance requirements. Notably, ALISE operates at the username level rather than the numeric user ID level, as UID resolution is handled transparently by standard operating system mechanisms (such as NSS or LDAP lookups).

ALISE currently supports OpenID Connect (OIDC) [7] for interfacing with local computer centre authentication systems. This protocol choice provides several advantages: users experience familiar login interfaces consistent with other institutional services, ALISE never handles user passwords directly, and the system can leverage existing institutional authentication infrastructure. Future releases will expand protocol support to include LDAP and SAML, providing greater flexibility for diverse institutional environments.

Following successful primary authentication, users can link multiple *external* federated identities to their local account. The ALISE web interface presents a curated list of supported identity providers, each accessible through a single click that initiates the appropriate authentication flow. This streamlined approach minimizes user confusion while maintaining security through standardized OIDC authentication protocols.

ALISE's federated identity support is built around AARC-BPA [3] compatible Community AAI systems, including Helmholtz-AAI and EGI-Checkin. This architecture enables users to link identities from their home institutions via eduGAIN [8], research-specific identifiers such as ORCID, and social identity providers including Google and GitHub. Direct Google account linking is also supported for demonstration and testing purposes.

The interface provides clear visual indicators of linkage status for each supported identity provider, allowing users to understand their current configuration at a glance. Users can also remove previously established linkages when they are no longer needed, providing full control over their identity mapping configuration.

4. Technical Platform and Architecture

ALISE is architected as a modern web application that provides both user-facing interfaces and programmatic API access. The system is implemented in Python 3, leveraging the FastAPI [9] framework for web services and the social-core [10] library for authentication provider integration. This technology stack was chosen for its robustness, active community support, and extensive ecosystem of authentication providers.

A key architectural challenge addressed by ALISE involves session management for multiple concurrent identity provider relationships. Standard authentication frameworks typically manage session state autonomously, with each new authentication flow resetting the session cookie. This behaviour limits the framework to maintaining a single active login with any remote identity provider. To enable the multi-provider linking workflow that ALISE requires, the system implements custom session management using dedicated session cookies that persist across multiple authentication sequences.

ALISE implements a multi-tenant architecture that enables a single deployment to serve multiple computer centres simultaneously. This design provides significant operational efficiency while maintaining clear isolation between tenants. From a user perspective, multi-tenancy requires an additional navigation step to select their target computer centre, though this can be streamlined by providing institution-specific entry points that bypass the selection interface.

The multi-tenant design is reflected in ALISE's URL structure, which incorporates the computer centre identifier as a path prefix (e.g., `/vega/` for VEGA users). This approach ensures clear namespace separation and enables straightforward configuration management for different institutional requirements.

The current ALISE deployment supports multiple production computer centres including VEGA and KBFI, with KIT maintained for development and testing purposes. CESGA support is under active consideration for future inclusion. Additionally, PSNC operates their own independent ALISE instance, demonstrating the system's deployability across different institutional environments.

Data isolation in ALISE is achieved through a tenant-specific database architecture, with each computer centre's linkage data stored in a dedicated SQLite3 [11] database. This design provides strong data isolation guarantees while simplifying backup, migration, and data governance procedures. The per-tenant database approach also facilitates future scaling scenarios, such as distributing different tenants across separate ALISE instances when operational requirements demand it.

4.1 Deployment and Operational Considerations

ALISE deployment follows established patterns for Python web applications, utilizing Gunicorn [12] as the WSGI HTTP server. The combination of FastAPI and Gunicorn provides a robust foundation for production deployments, with proven scalability and reliability characteristics.

The primary scalability consideration in ALISE stems from the interaction between the social-auth library and the custom session management required for multi-provider authentication flows. This combination presents thread-safety challenges that are most reliably addressed by operating ALISE in single-threaded mode under systemd management. While this architectural constraint limits concurrent request processing, the expected workload characteristics make this acceptable: database writes occur only during initial account linking setup, while the majority of API requests are read-only mapping queries that perform efficiently even in single-threaded operation.

The deployment architecture employs Nginx [13] as a reverse proxy fronting the Gunicorn application server. This configuration follows well-established best practices for Python web applications and provides several operational benefits. Nginx handles SSL/TLS termination, certificate management, static file serving, and basic request filtering, while delegating application logic to the ALISE backend. This separation of concerns improves security by leveraging Nginx's proven track record in handling web security threats and reduces the attack surface of the ALISE application itself.

4.2 Configuration and Customization

ALISE's configuration system is designed to accommodate the diverse requirements of different computer centres while maintaining operational simplicity. The configuration approach recognizes

that different OpenID Connect providers (OPs) may vary in their attribute schemas, claim names, and available scopes. For internal institutional OPs, ALISE provides flexible mapping capabilities, allowing administrators to specify which OIDC claim contains the local username information.

The configuration for each identity provider is defined through concise configuration sections that capture all necessary integration parameters. The following example demonstrates the configuration structure for a typical institutional provider:

```
[auth.vega]
# redirect-uri: https://alise.data.kit.edu/oauth2/vega/token
op_url = https://sso.sling.si:8443/auth/realms/SLING
client_id = alise-prod
client_secret = thequickbrownfoxjumpsoverthelazydog
scopes = openid profile email roles acr
username_claim = upn
internal = True
```

Beyond standard OIDC parameters (client ID, client secret, and OP URL), ALISE's configuration supports several specialized settings that enable integration with diverse institutional environments. The `scopes` parameter allows customization of the OIDC scope request to obtain appropriate user attributes. The `username_claim` setting specifies which OIDC claim contains the local username, accommodating variations in institutional attribute schemas. The `internal` flag distinguishes between local institutional providers and external federated providers, enabling appropriate security and governance policies.

The redirect URI configuration must align with the local OP configuration name, ensuring proper callback routing during the authentication flow. Additionally, administrators must provide appropriate iconography for each computer centre to maintain consistent branding in the user interface.

5. Developer Interface: The ALISE API

The ALISE REST API provides application developers with programmatic access to the identity mapping database, enabling seamless integration of federated identity resolution into existing services and workflows. The API design emphasizes security, auditability, and ease of integration while maintaining strict access controls over sensitive identity mapping data.

Access to the ALISE API requires authentication via API keys, which can be generated by any user who has successfully authenticated with a supported identity provider. This approach ensures that only legitimate users can access mapping services while providing a scalable mechanism for service integration. The system maintains comprehensive audit logs of all API accesses, enabling administrators to trace every data access back to the originating user and facilitating compliance with data protection regulations.

Future enhancements may include support for fine-grained authorization through AARC-style entitlements, enabling more sophisticated access control policies based on user roles or institutional affiliations.

The ALISE API implements a RESTful interface with the following core endpoints, all of which return structured data in JSON format to facilitate easy integration with modern application development frameworks:

- `/api/v1/version`: To obtain the current version of ALISE
- `/api/v1/alise/supported_issuers`: Returns the supported issuers
- `/api/v1/target/<site>/get_apikey`: Requires an access token issued by a supported issuer. Returns an API-key
- `/api/v1/target/<site>/validate_apikey/<apikey>`: Validates an API-key, returns `true` or `false`
- `/api/v1/target/<site>/mapping/issuer/<encoded_iss>/user/<encoded_sub>?apikey=<apikey>`: Returns all mappings of a user identified by its `sub` and `iss`.

These API endpoints enable federated services to perform real-time identity resolution, translating federated identity assertions into the corresponding local Unix accounts required for traditional computing infrastructure. The API's stateless design ensures high performance and scalability for production service integration.

5.1 API Usage Example

To illustrate the practical application of the ALISE API, consider a scenario where a service needs to resolve the Unix account corresponding to a federated user. The following example demonstrates retrieving mappings for a user identified by the OIDC claims `sub=42234223-4223-4223-4223-422342234223` and `iss=https://login.helmholtz.de/oauth2`:

```
curl  https://alise.data.kit.edu/api/v1/target/vega-kc/mapping\
      /issuer/$(urlencode.py https%3A%2F%2Flogin.helmholtz.de%2Foauth2)\
      /user/$(urlencode.py 42234223-4223-4223-4223-422342234223)\
      ?apikey=inyourdreamsinyourdreams | jq
```

The API returns a comprehensive JSON structure containing both the internal account information and all associated external federated identities:

```
{
  "internal": {
    "sub": "4223422-42234-4223-4223-42234223",
    "iss": "https://sso.sling.si:8443/auth/realms/SLING",
    "username": "marcush",
    "last_seen": -930016800,
    "display_name": "Marcus Hardt"
  },
```

```
  "external": [
    {
      "sub": "42234223422342234223422342234223422342234223422342234223422336@egi.eu",
      "iss": "https://aai-demo.egi.eu/auth/realms/egi",
      "last_seen": 13552005600,
      "display_name": "Marcus Hardt"
    },
    {
      "sub": "42342234223422342234",
      "iss": "https://accounts.google.com/",
      "last_seen": 13552005600,
      "display_name": "Marcus H"
    },
    {
      "sub": "42234223-4223-4223-4223-422342234223",
      "iss": "https://login.helmholtz.de/oauth2",
      "last_seen": 13552005600,
      "display_name": "Marcus Hardt"
    }
  ]
}
```


The `last_seen` timestamps provide crucial temporal information that services can use to make informed authorization decisions based on the recency of identity verification. Services requiring fresh affiliation information (such as verifying current faculty status) may enforce stricter temporal requirements than those validating historical identity assertions (such as confirming assurance level compliance with `https://refeds.org/assurance/IAP/medium`). This flexibility enables services to implement appropriate risk management strategies based on their specific security and compliance requirements.

6. Conclusions and Impact

This paper has presented ALISE, the Account LInking SErvice, which addresses a critical infrastructure gap in modern research computing environments. By enabling seamless mapping between federated identities and local Unix accounts, ALISE bridges the architectural divide between contemporary federated authentication systems and traditional compute centre infrastructure.

Our comprehensive examination of ALISE from multiple stakeholder perspectives demonstrates the system's effectiveness in addressing the diverse requirements of users, developers, and system administrators. The user-centric design simplifies the account linking process while maintaining strong security guarantees. The developer-focused API enables straightforward integration with existing services and applications. The administrative interface provides operators with the tools necessary to deploy and maintain ALISE in complex institutional environments.

ALISE's deployment across multiple production computer centres validates its practical utility and demonstrates the scalability of the multi-tenant architecture. The system successfully addresses

real-world requirements for integrating federated authentication with traditional Unix-based infrastructure, enabling institutions to modernize their authentication workflows without abandoning existing computational investments.

6.1 Future Directions and Development Roadmap

ALISE continues to evolve through active development driven by community requirements and emerging standards in federated identity management. Several significant enhancements are planned to expand the system's capabilities and broaden its applicability.

Near-term development priorities include integration with ssh-oidc [14] through the motley-cue [15] framework. This integration will enable ssh-oidc deployments to leverage ALISE's identity mapping database, creating a unified approach to federated identity resolution across multiple access modalities including SSH, web services, and storage systems.

Protocol expansion represents another key development area, with planned support for LDAP and SAML authentication protocols. These additions will enable ALISE deployment in environments where OpenID Connect is not the primary institutional authentication protocol, broadening the system's applicability across diverse institutional technology stacks.

A particularly innovative feature under development involves group-based account mapping, where multiple federated users can be mapped to shared Unix accounts based on structured entitlements. This capability will leverage AARC-G069 [16] entitlement specifications to enable sophisticated authorization policies that support collaborative computing scenarios and project-based access models.

The open source nature of ALISE (MIT License) facilitates community contributions and collaborative development. The project welcomes community involvement through its GitHub repository [1], enabling researchers and developers to contribute enhancements, bug fixes, and additional features that benefit the broader research computing community.

References

[1] Account LInking SErvice – ALISE (Online; accessed 17-April-2024) https://github.com/m-team-kit/alise

[2] WebDAV network protocol for remote collaborative authoring on the Web https://link.springer.com/chapter/10.1007/978-94-011-4441-4_16

[3] AARC Blueprint Architecture 2019 version https://zenodo.org/doi/10.5281/zenodo.3672784

[4] ORCID, the free, unique, persistent identifier (PID) for individuals (Online; accessed 17-April-2025) https://orcid.org/

[5] AARC-G031 Guidelines for the evaluation and combination of the assurance of external identities https://aarc-community.org/guidelines/aarc-g031/

[6] AARC-G025 Guidelines for expressing affiliation information https://aarc-community.org/guidelines/aarc-g025/

[7] OpenID Connect (OIDC) https://openid.net/developers/how-connect-works

[8] eduGAIN international interfederation service https://edugain.org

[9] FastAPI web framework for building APIs with Python based on standard Python type hints https://fastapi.tiangolo.com/

[10] Social Core, a python library to support a large set of identity providers https://github.com/python-social-auth/social-core

[11] Sqlite, a small and efficient SQL database engine https://www.sqlite.org

[12] Gunicorn is a Python WSGI HTTP Server for UNIX https://gunicorn.org/

[13] nginx ("engine x") is an HTTP web server, reverse proxy, content cache, load balancer, TCP/UDP proxy server, and mail proxy server. https://nginx.org/en/

[14] Enabling Secure Shell Access with OpenID Connect https://link.springer.com/article/10.1007/s41781-025-00136-5

[15] motley-cue https://zenodo.org/doi/10.5281/zenodo.7346725

[16] AARC-G069 Guidelines for expressing group membership and role information https://aarc-community.org/guidelines/aarc-g069/